GIT
CLASE 3 : FLUJOS DE TRABAJO

EJERCICIOS :

1. Vamos a seguir trabajando sobre el repositorio original que hayamos generado en los ejercicios de la clase 1. En este caso, si no lo hicimos aún, vamos a actualizar nuestro branch master para que coincida con nuestro branch desarrollo, es decir que ambos queden en el mismo nivel.
2. Tenemos que crear tres branches nuevos que salgan desde el branch desarrollo (B1,B2,B3 o pueden usar el nombre que quieran, como por ejemplo frontend, backend,qa, etc).
3. Vamos a ir ingresando a cada branch e ir creando un archivo nuevo en cada uno, de manera que cuando nos pasemos al siguiente o anterior branch, ese archivo desaparezca.
4. Le vamos a ir haciendo uno o más cambios al archivo de su respectivo branch y vamos a mostrar en consola el historial de nuestros commits. Deberíamos notar que tenemos tres branches con historial bifurcado desde el branch desarrollo.
5. Vamos a elegir dos de estos branches, por ejemplo B1 y B2 y vamos a juntar sus cambios utilizando el método merge. De esta manera los cambios de ambos van a estar incorporados en una sola rama de commits.
6. Ahora nos va a quedar el branch B3 sin incorporarse a nuestros otros dos branches B1 y B2. En este caso lo vamos a hacer pero utilizando el método de rebase.
7. Por último vamos a incorporar todos los cambios a nuestro branch de desarrollo para que este esté actualizado con todos los últimos cambios generados y vamos a subir todo a nuestro repositorio en GitHub.

BONUS :

1. Primero que nada vamos a generar una versión oficial de nuestro proyecto generando un tag. Para esto vamos a necesitar actualizar nuestro branch master para que el mismo incorpore los cambios que tenga desarrollo.
2. Una vez que tengamos nuestro branch master actualizado y nuestro tag generado, vamos a subir el mismo utilizando el método push.
3. Crear un branch desde desarrollo el cual lo vamos a utilizar solo para test. En el mismo, vamos a realizar 5 cambios distintos y guardarlos en 5 commits distintos.
4. Utilizando el comando bisect, vamos a intentar borrar alguno de esos cambios y el resto los vamos a unificar en un solo commit. Si estos cambios no influyen demasiado en nuestro proyecto, podemos unificarlos con nuestro branch desarrollo y sucesivamente con el master, generando un nuevo tag y subiendo todo a GitHub.